CDF Tracking Experiance

Matthew Herndon
Johns Hopkins University

CMS Tracking Workshop

August 3, 2004

Outline

General Introduction to CDF Tracking

Infrastructure and Tools for Tracking Development

Getting Ready for Data Taking

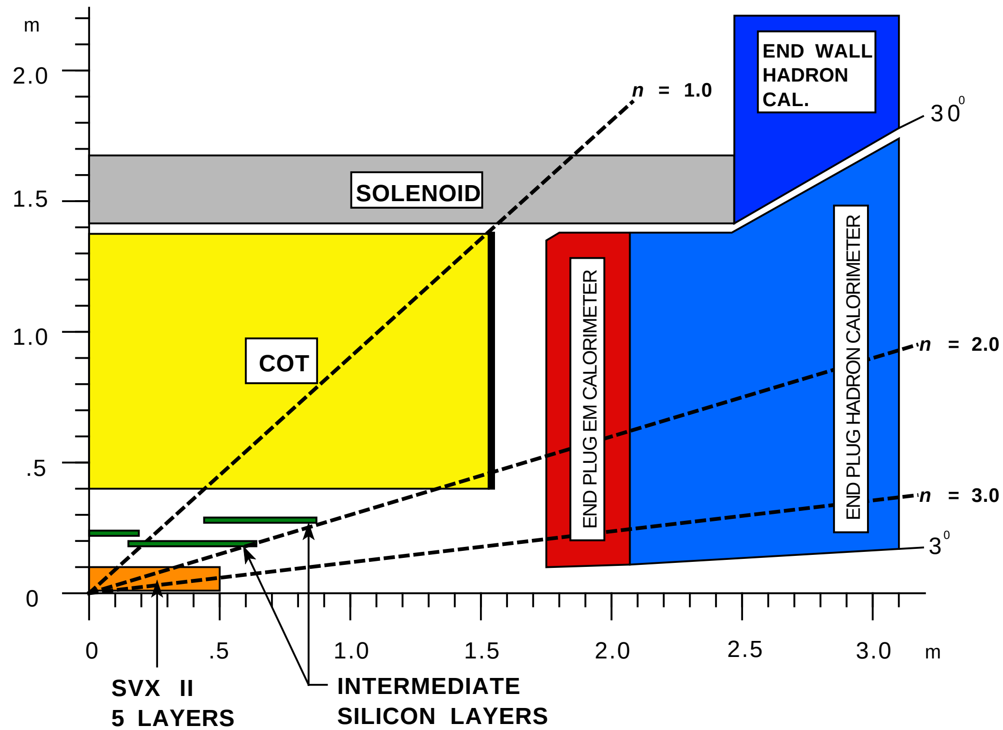
Conclusions

Run 2 Tracker

Drift chamber + silicon detector

- Central outer tracker: COT
  - 8 super layers: 4 axial 4 stereo
    96 measurements between
    44 and 132cm
  - Tracking for $|\eta| < 1.0$
  - Axial layers used for level 1 trigger
- Resolution
  - Hit resolution 140μm
  - $\delta \mathrm{p_T}/\mathrm{p_T^2} \sim 0.15\% (\mathrm{GeV/c})^{-1}$
  - Impact parameter resolution of 250μm
  - Z0 resolution of 5mm
- Silicon detector
  - 8 layers(7 double sided)
  - Tracking for $|\eta| < 2.0$
    with two outer layers

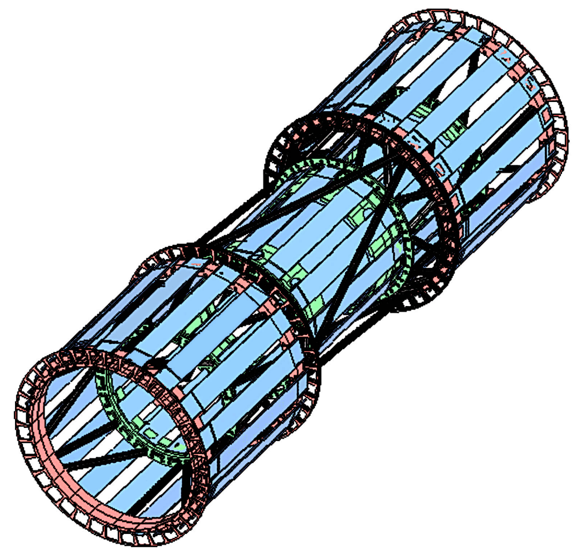
CDF Tracking Volume

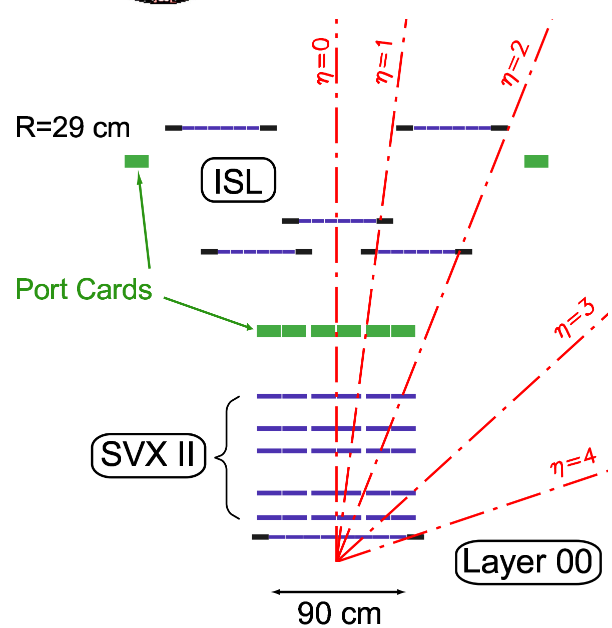


Run 2 Silicon

Features

- 8 layers, 7 stereo sided
  3 90° Z and 4 small angle
- Standalone tracking for $|\eta| < 2.0$
  - Silicon only tracking at a hadron collider
- 3D tracking - 3 layers of 90° strips
- L00
  - Precision measurement at $1.3 - 1.6\mathrm{cm}$
  - High density environment
- Resolution
  - Impact parameter resolution of 25μm ($\sim 40\mu\mathrm{m}$ with beam line)
  - Z0 resolution of 50μm
- Used in second level trigger
  - Displaced track trigger
  - Cuts at 120μm with 35(47)μm res





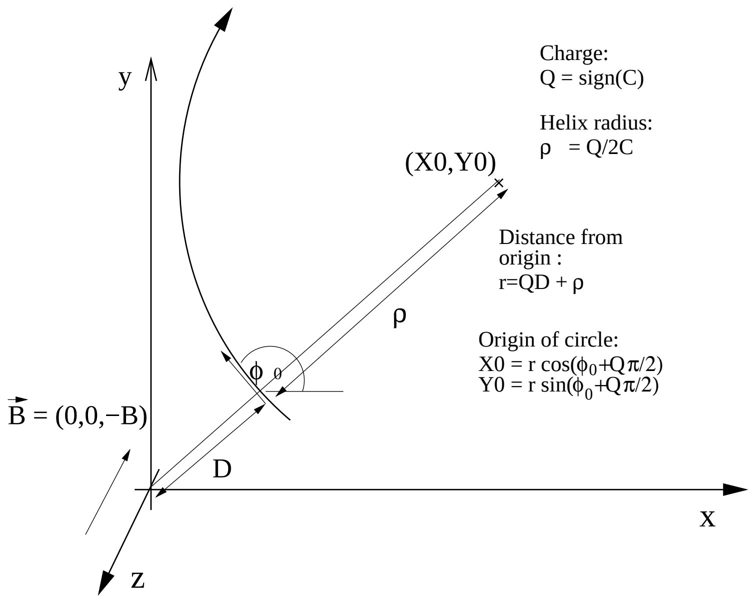
CDF Coordinates

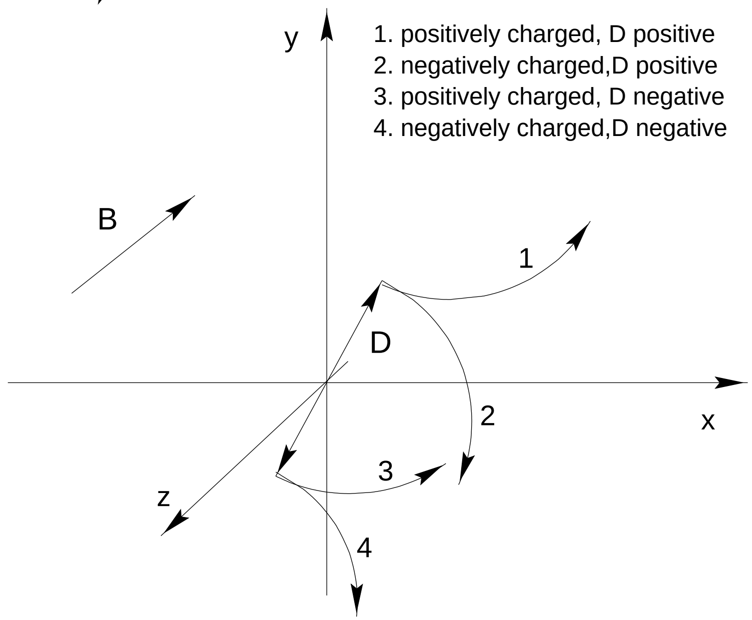
Perigee track parameterization

- $\cot\theta$: cotangent of the polar angle at minimum approach to the origin in $\mathrm{r}-\phi$
- C: half curvature (signed by charge)
- z0: z position at min
- d0: impact parameter at min (signed by charge and curvature)
- ϕ_0 phi at min

Useful parameterization for physics

- Usually use pt in physics analysis
- pt, ϕ_0 and $\cot\theta$ give mass resolution
- d0 and z0 give vertexing resolution

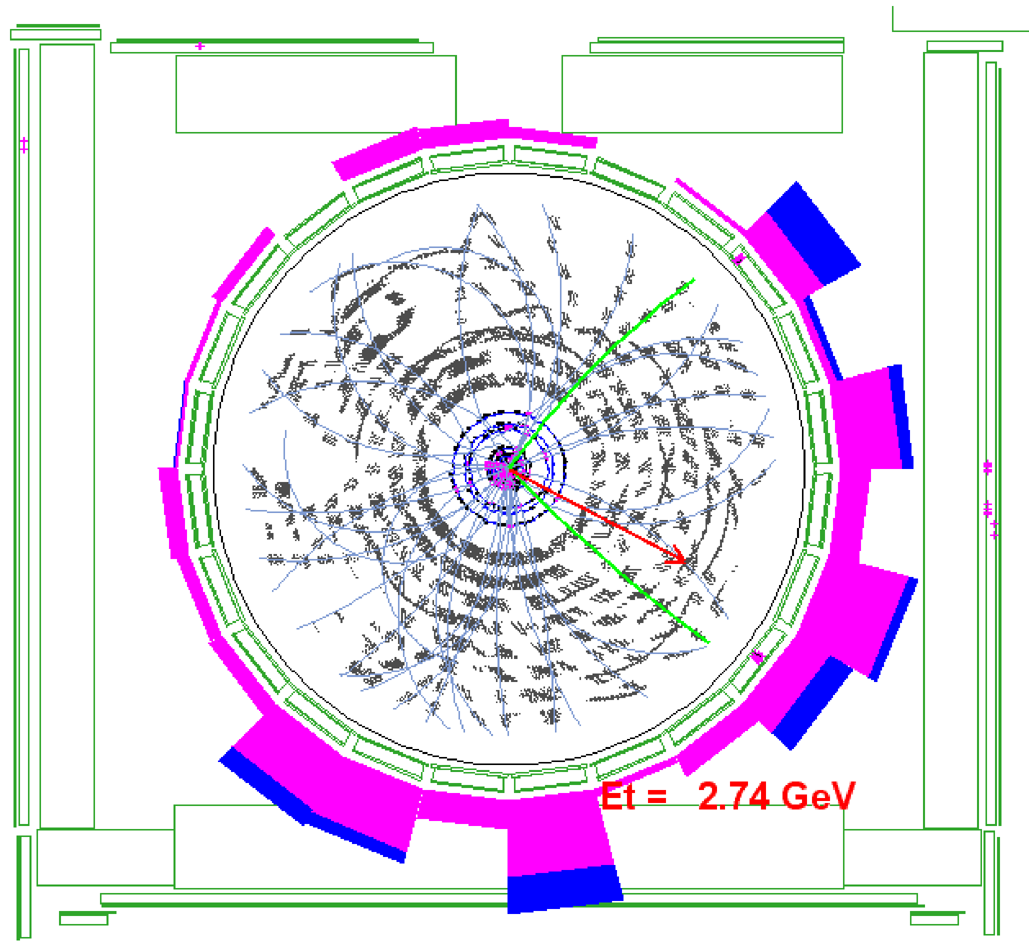




General Concepts of Tracking

Procedure

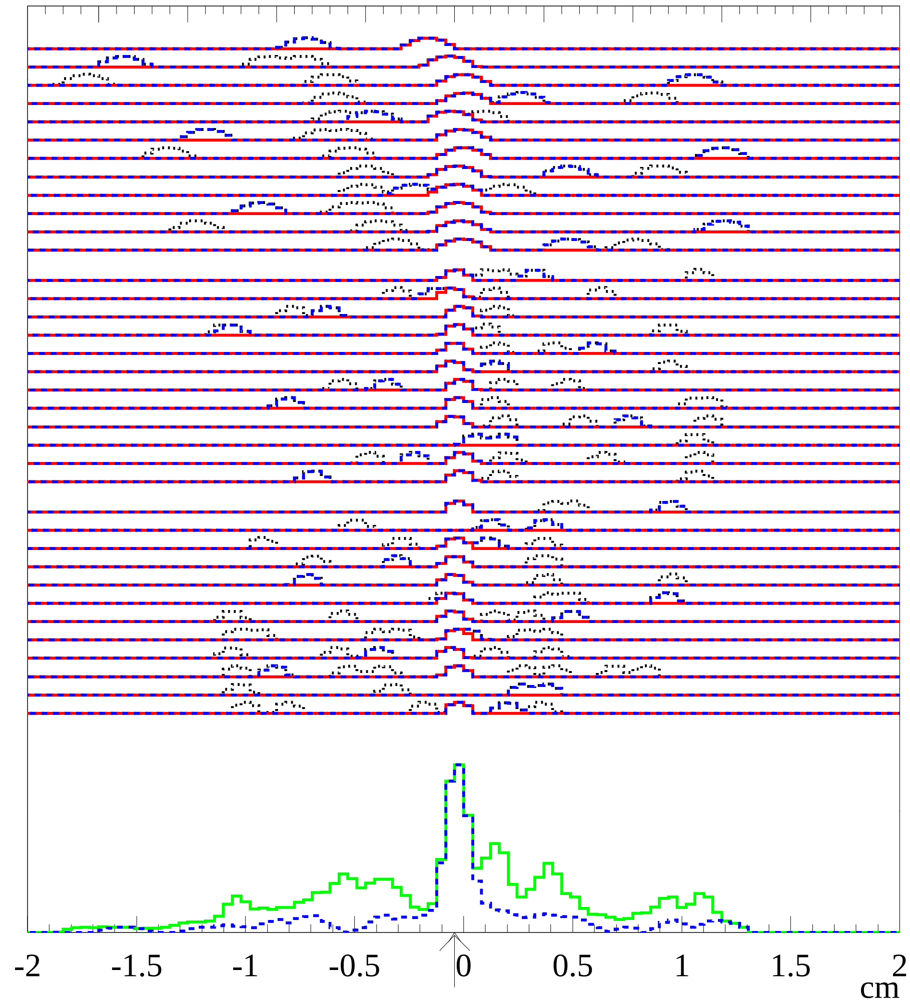
- Start where evidence of tracks is the most obvious
  - Outermost radius of COT or silicon layers 6,7
  - Actually angular resolution of COT and silicon are similar at their outermost radii
- General method: Outside in search
  - Find evidence of tracks in the outer radii
  - Form preliminary tracks
  - Look for consistent hits further in
  - Repeat
- Silicon only tracks
  - Silicon has a maximum of 8 measurements(typically 5-6) COT has 96 measurement points
  - Considerably more difficult pattern recognition problem



COT Tracking Algorithms

COT Algorithms

- Start with segments in the outer SLs
  - High probability of being from a track
- Segment linking algorithm
  - Link segments by curvature matching
- Histogram linking algorithm
  - Use segment curvature and beamline to form road
  - Add hits using histograming method
  - Fast method suitable for low mass detectors: minimal multiple scattering
    Large scattering distorts the histogram
- Complementary methods
  - Merge tracks from two methods
  - Two methods help increase efficiency and reduces fake track rate(5% gain)

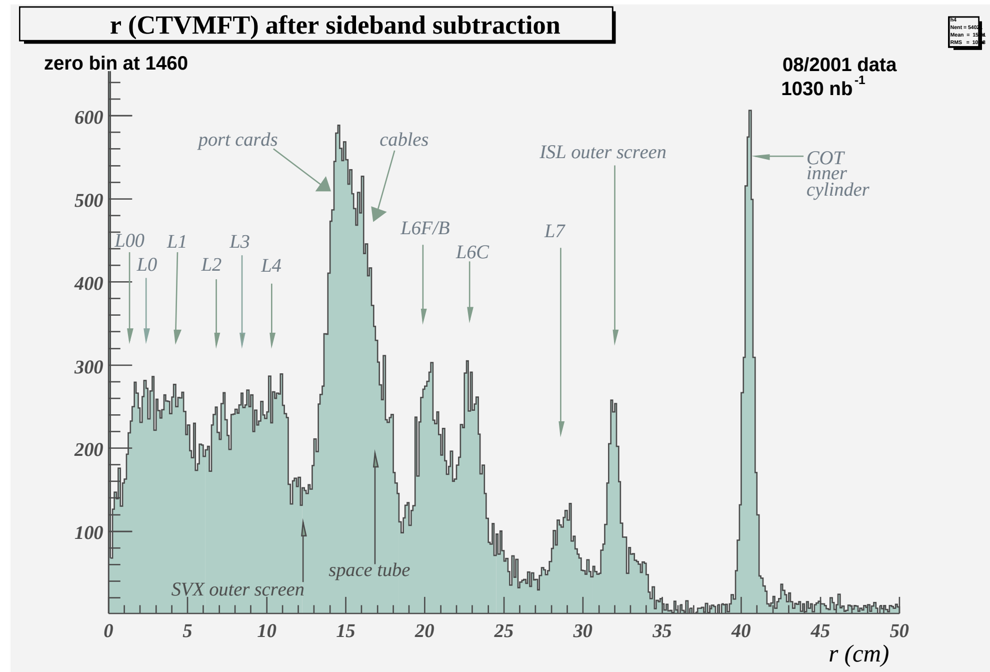


Histogram formed relative to the nominal search path

Silicon Algorithms

Central silicon tracking

- Outside-In tracking
  - Uses COT track as seed
  - Progressive road search
  - Fitting performed using a fast kalman filter approach
  - Excellent for progressively incorporating discreet measurement points with errors
  - Tracking done in 2 passes: r-phi and r-z
  - OI silicon reconstruction fast enough to be used in level 3 trigger
- Note: tried a histograming method
  - Histograms too distorted to be useful
  - Not very fast:
    Had to make histogram wide to account for scattering and then there are too many hits

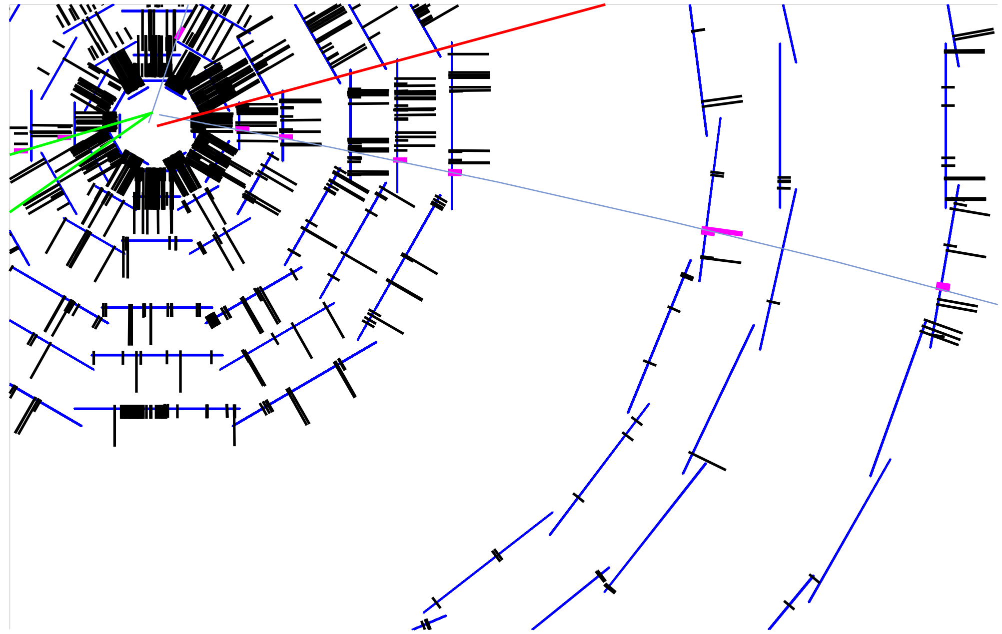


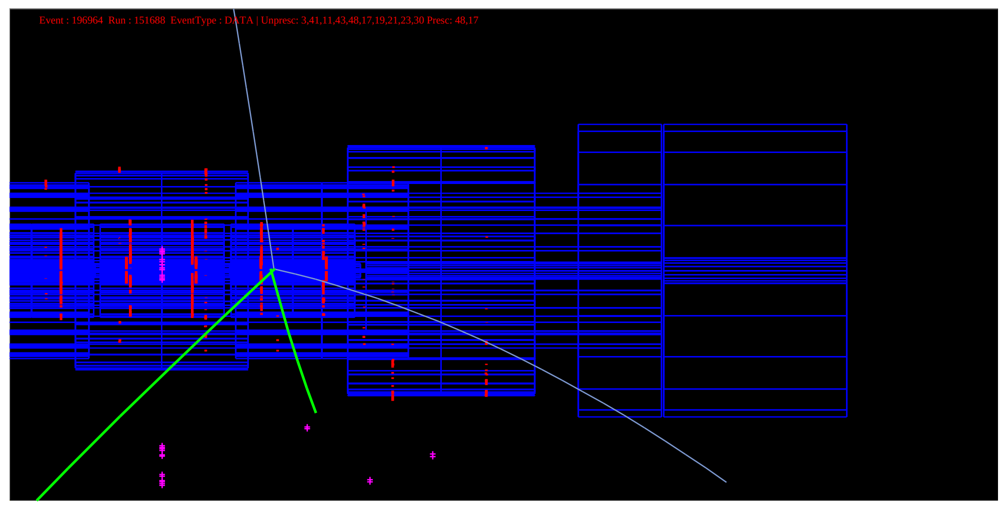
Photon conversions in the silicon detector - Kalman filter used in conjunction with material integrator

Silicon Algorithms: Cont

Silicon only tracking

- Standalone silicon tracking
  - Tracking outside of acceptance of COT
  - Start with two 3D measurement points on the outer silicon layers
  - Road from the points and the beamline
    3 points form a helix
    Low prob of being a track
  - Progressive road search as in OI case (uses same fitter)
  - Methods using less information had a very high fake track rate
    1 3D and 1 2D hit, 1 3D hit
- Methods to reduce combinatorics
  - Look at track candidates that use hits not used by OI algorithm
  - Require that track be compatible in Z with vertex found by OI tracks





Tracking Infrastructure

Basic Objects

- Detector hits and tracks
  - Each should have a simple base class
    Fast version that interacts with fits - essentially just measurement points or a helix and errors
  - High level version with all the information
    Links to hits, write, read and print methods: Would have been good to have muon and calor links
- Other objects: Beamlines, vertices - Also measurement points

Actions on basic objects

- Track fits, material integrators, and extrapolators
  - Likely to need several versions: fast versions for pattern recognition
  - Not too generalized if it adds complexity
    For instance, having this infrastructure be common with the muon system wasn't found to be useful. The track extropalator was much slower than a simpler parameterized version
    Inheritance and templating only useful if they improve performance or maintainability

Simulation

- Integrate objects with the simulated versions from the beginning
  - Simulation processing should be identical to data processing
  - Geometry and other infrastructure should be in common

Simple Tools for Tracking Development

Print methods

- Very useful for debugging
- In addition to the methods for hits and tracks - also for geometry, raw data formats, simulation information, internal fit objects...

Simple ntuples

- Ntuples at the track and hit level
  - All track and hit parameters and covariance matrix elements
- Can evolve over time to include more information
  - d0 and z0, relative to a beamline or z vertex
  - Residuals to hits
  - Hit efficiency information: which hits were expected and found
  - Comparisons to simulation data
  - Comparisons between different algorithms
  - Comparisons to jet, calorimeter or muon information

Advanced Tools for Tracking Development

Print methods -Somehow debugging always comes back to this

Reconstructed and Simulated track association

- Helix parameter association - match helix parameters
  - Fairly easy to write
  - Tend to see discrepancies when there is large scattering or energy loss

- Hit level association - match according to the hit content of the tracks
  - More powerful method
  - All the hit information is immediately available - calculate hit efficiencies and fake hit rates
  - Can fit the hits to find best possible track - Perfect Tracking
    Make sure that track fitting can be driven off a list of hits - Unified data and simulation processing

Reconstructed - Reconstructed track association

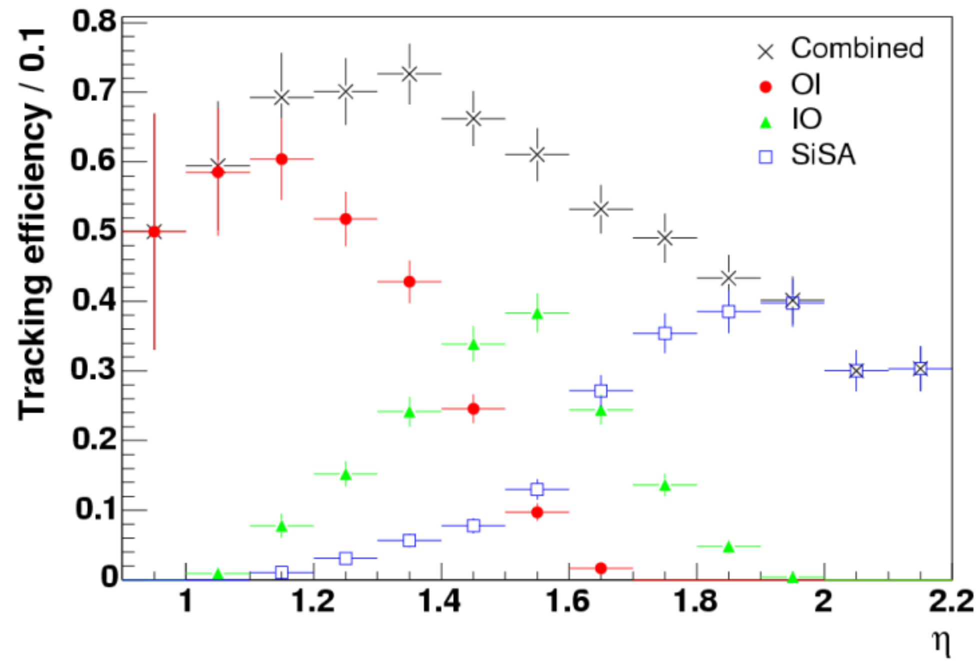
- From the standpoint of hit level association comparing reconstructed track from different algorithms should be similar to comparing reconstructed and simulated tracks

Add all to the ntuple: CDF association information was in a separate ntuple

Tools for Tracking Data Analysis

Tracking Efficiency

- Define performance based on data quantities
  - Efficiency from $Z \rightarrow ee$
  - One identified electron track and a calorimeter energy deposit consistent with a Z mass
- Used for forward silicon efficiency
- Forward tracking still low efficiency
  - Many forward ladders dead (cooling doesn't work)
  - Difficult to align forward part of detector
  - Solved many problems when we decided to concentrate isolated high pt tracks



Important to make algorithms robust against problems

No need to wait for data to write these tools

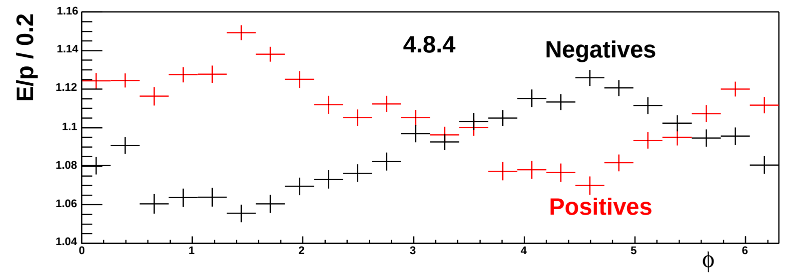
Tools for Tracking Data Analysis: Resolution

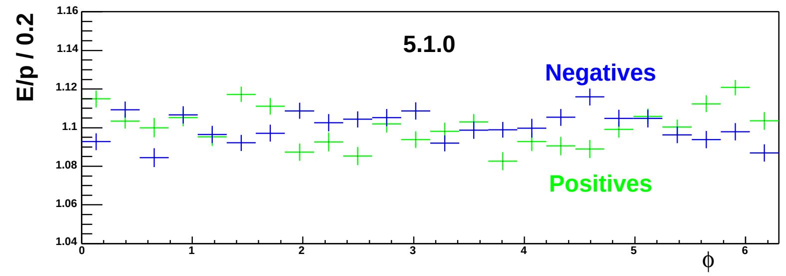
Curvature, ϕ_0 and $\cot\theta$ resolution

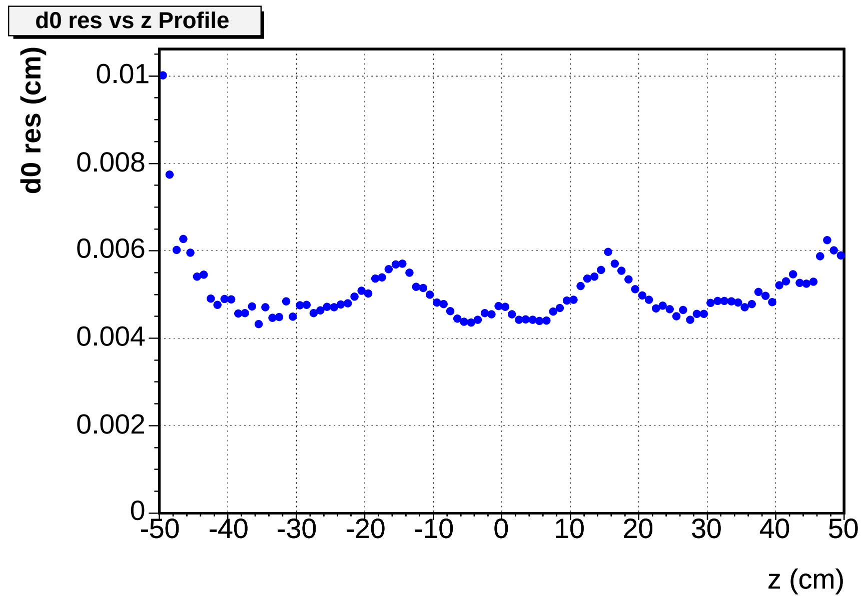
- E/p resolution compared to the calorimeter
  - Even if tracker has better resolution this is sensitive to systematic effects
  - We had an unexpected twisting of the COT end-plate
- Mass resolution for multi-body decays
  - Resolutions, material effects, field strength

Vertexing parameters: d_0 and z_0

- Profiles vs pt, ϕ, and Z
- Resolutions and pull distributions
  - Alignment problems, material, intrinsic detector resolution
  - Some of our fit problems were evident even in simulated data



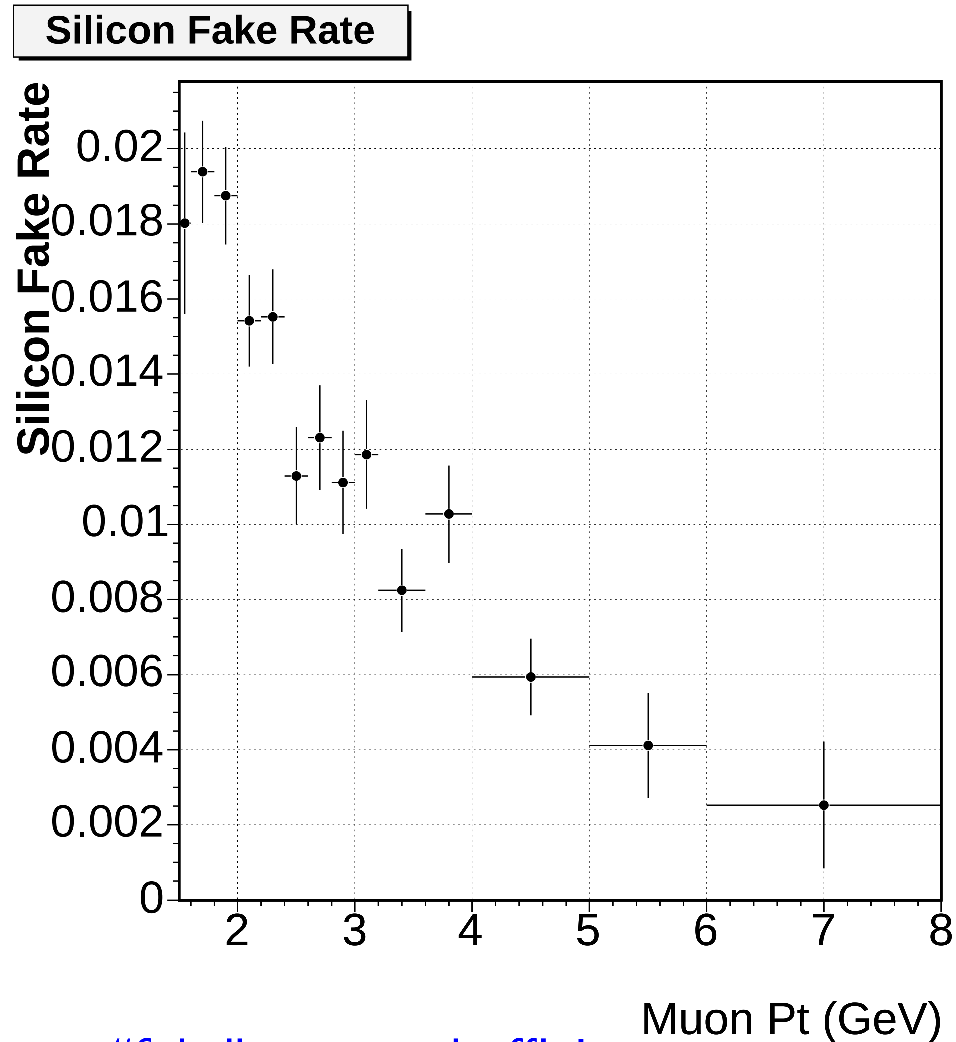




Tools for Tracking Data Analysis: "Fake" Rates

Rates of fake or mis-measured tracks

- Can be found from looking at the tails of distributions
- Events in the tails of the signed impact parameter distribution consistent with negative lifetime
  - Sign d_0 by jet or multibody decay momentum vector direction
  - Measure % below -3σ
  - CDF mis-measured track rate quite good
- Can also looks at z_0 and mass tails
- Fake tracks more difficult - often use simulation



Note: tuning often becomes a balance between "fake" rate and efficiency

Other Tools: Track Embedding

Embed tracks in real data

- Useful for determining efficiencies and studying resolutions
  - Efficiency of low momentum tracks difficult to measure in data
  - Simulation doesn't do a good job of reproducing the rates of low energy tracks or secondaries
  - Studies of L00 show a large number of very high eta and low pt tracks
- Data events can be used at any desired luminosity or track density
- Embed single tracks or whole physics processes in the events
- Some tuning usually necessary to reproduce distributions like hit resolutions
- Can often very effectively describe the data when used with a good simulation
  - Number of hits on tracks
  - Width and tails of resolution distributions
  - Track efficiencies
- Infrastructure can be made compatible with that for mixing simulated min bias and hard scattering events
  - Always wanted to run some events through Geant with very low energy cut offs - might do a good job describing occupancy on inner detector components

Other Problems: Using Real Data

Considerable problems can be encountered reading real data

- Bit error(CMS uses optical readout also)
- Undocumented features of the data format
- Confusion about detector geometry
- Should set up tests of reading real data as soon a possible
  - We created a wedge in a box which gave us a head start on all these problems
  - Went to the level of finding cosmic ray tracks

Getting the data

- Need quick access to useful physics sample
  - Generalized tracks for analysis: J/Psis, Zs, high pt electrons and muons
  - Also the processing power allocated to quickly reprocess them
    Turn around times of hours in order to test new tracking ideas
  - True even now for tests on simulation

Other Problems: Alignment

Probably the single biggest tracking project

- Very difficult and time consuming
- Can be refined almost indefinitely: we are still improving the CDF alignment
- Considerable preparation should be done
- Realistic Alignment tests
  - Include the usual random misalignments
  - Also include systematic effects:
    radial displacements of whole layers, rotations(seems likely at CMS with the tilted layer design), rotations of layers with respect to each other, translations in xzy especially between separate parts of the detector
  - What did we see at CDF
    All of the above and the stereo angle of our double sided sensors turned out to be variable
  - Once we began to understand what problems could exist these alignment test modes were useful in fixing the problems in addition to validating the alignment procedure
- Also the ability to load in and compare to construction survey data
- Another reason to make the algorithms robust against various problems
  - Wider then normal search windows can be used. Should be easily tunable

Validation

Validation procedures

- Should set up validation procedures early
- Run with every new release - can be automated eventually
- Test all tracking performance parameters
  - Functionality of the code: tracking runs, tools all run, ntuples made
  - Efficiency of algorithms
  - Resolution and pull distributions of all track parameters
  - Test both your simulation based and your data like tools
  - Current speed of the code - optimization is often needed
    Note that the code that extensively used templating and inheritance was difficult to optimize
    (one tracking algo was discarded as hopeless)
- Also check various code development issues
  - Crashes when running over large datasets
  - Memory leaks and other memory issues: using tools like valgrind
  - Fixing these issues will speed up development
- Easy to make backward steps while developing and tuning the code

Physics

Use the physics groups

- Once data taking has started most of the man power is in the physics groups
- No problem: Physics is the best tool to improve tracking
  - Besides you get to do physics at the same time
- Physics analyses that critically depend on the tracking can be used to improve the tracking performance
  - Exotics analyses will depend on lepton finding efficiencies
  - B tagging based analyses can be used to improve the understanding of tails and resolutions
  - Encourage and direct people in performing low level studies
- Migrate tools to monitor tracking into physics analysis.
  - This will make sure you keep track of what's most important and help to spot problems that were not obvious
  - Keep all the more basic tools around for understanding problems once identified

Conclusions

Tried to share some of the CDF experience in tracking

- CMS should concentrate on building a set of tools for tracking development
- These tools can be used to improve the code and prepare for real data taking
- The LHC will be very exciting with the potential to be seeing the signs of new physics very quickly! The better the tracking works on day 1 the sooner that may happen.
- I'm willing to help as soon as you offer me a job :)